

# Practical Malware Analysis & Triage

## Malware Analysis Report

### WannaCry Ransomware Cryptoworm

December 2021 | An00bRektn | v1.0

# Table of Contents

Table of Contents .....	2
Executive Summary.....	3
High-Level Technical Summary .....	4
Malware Composition.....	5
wannacry.exe .....	5
tasksche.exe .....	5
Additional Files.....	5
Basic Static Analysis.....	7
Preliminary Information Gathering .....	7
Strings .....	7
PEStudio.....	9
Basic Dynamic Analysis.....	11
Initial Detonation – With Internet Simulation .....	11
Initial Detonation – Without Internet Simulation.....	12
Advanced Static Analysis.....	17
Advanced Dynamic Analysis.....	18
Indicators of Compromise .....	19
Network Indicators .....	19
Host-based Indicators .....	20
Rules & Signatures.....	21
Appendices.....	22
A. Yara Rules .....	22
B. Disassembled Code Snippets .....	23

# Executive Summary

SHA256 hash	24d004a104d4d54034dbcfc2a4b19a11f39008a575aa614ea04703480b1022c
-------------	---

WannaCry is a ransomware sample first identified on May 12<sup>th</sup>, 2017. It is a C++-compiled cryptoworm that runs on the x64 and x86 Windows operating systems. The sample consists of a main payload that unpacks an additional payload, along with helper files, to encrypt files on a host and demand a ransom for decryption. WannaCry propagates across a network by leveraging a known vulnerability in the Windows operating system to give full Administrator access, that has since been patched by Microsoft.

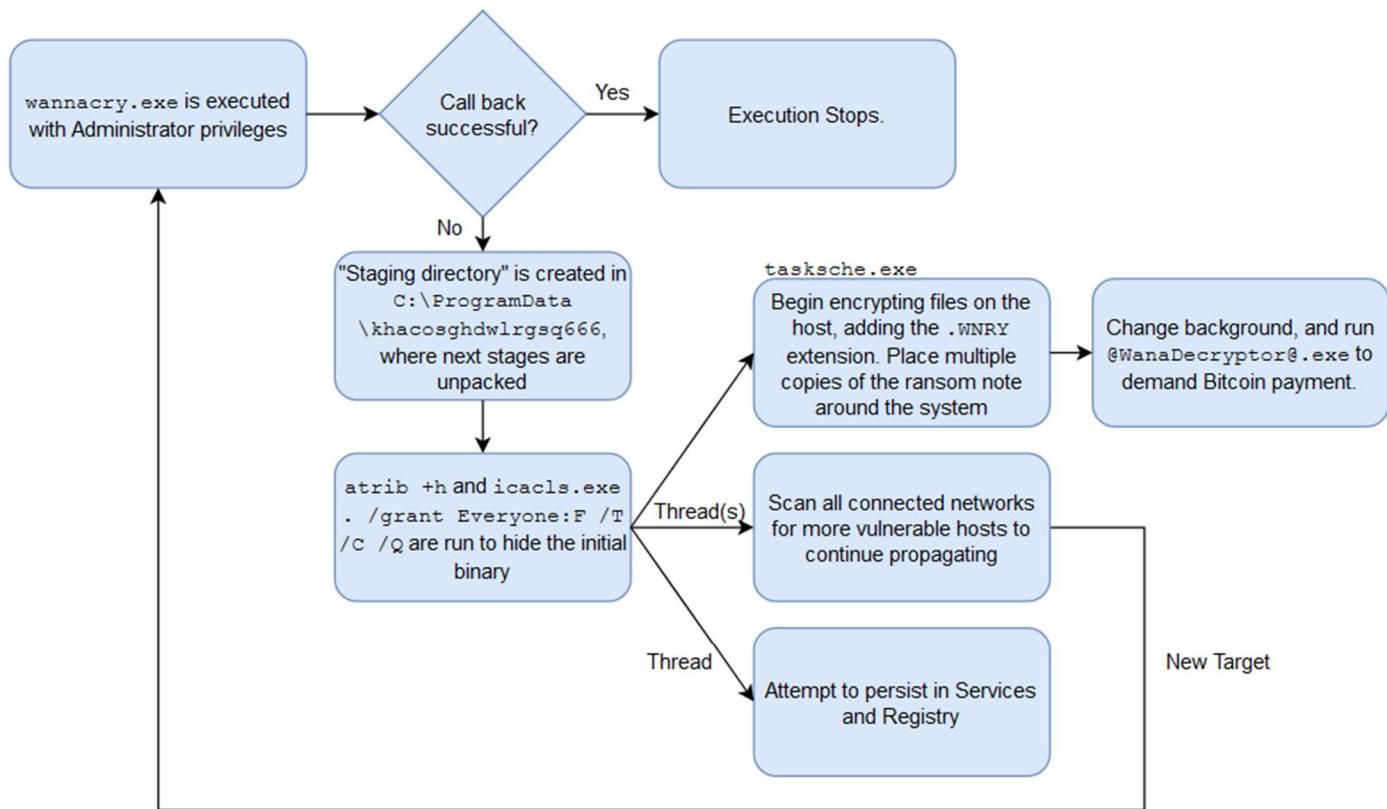
Symptoms of infection include, but are not limited to:

- Files with the ‘.WNRY’ extension and a visibly different wallpaper saying that files have been encrypted
- an executable named “@WanaDecryptor@.exe” asking for payment to a given Bitcoin wallet
- A new, hidden directory with a name of random characters located in C:\ProgramData\
- Creation of a new service with the same name as the hidden folder in C:\ProgramData\ used for persistence
- Shadow copies of the file system removed from disk
- A taskhsvc.exe executable listening on port 9050

YARA signature rules are attached in Appendix A. Malware sample and hashes have been submitted to VirusTotal for further examination.

## High-Level Technical Summary

WannaCry consists of a main payload that attempts to contact its callback URL (`hxxp://www.iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea.com`). If the call back is **unsuccessful**, it unpacks its next stage and other “helper” files into `C:\ProgramData\[RANDOM_STRING]`. The flow of execution is shown in the diagram below.



*Fig 1: Flowchart depicting the execution of WannaCry.*

WannaCry leverages MS17-010 (dubbed “EternalBlue”) to propagate across the network, which abuses a vulnerability in the Windows SMBv1 protocol. A patch from Microsoft had already been released prior to the initial discovery of WannaCry, however, hosts that have not applied the vendor patch are susceptible to the sample.

# Malware Composition

WannaCry consists of the following stages:

File Name	SHA256 Hash
wannacry.exe	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
tasksche.exe	ed01ebfb9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

## wannacry.exe

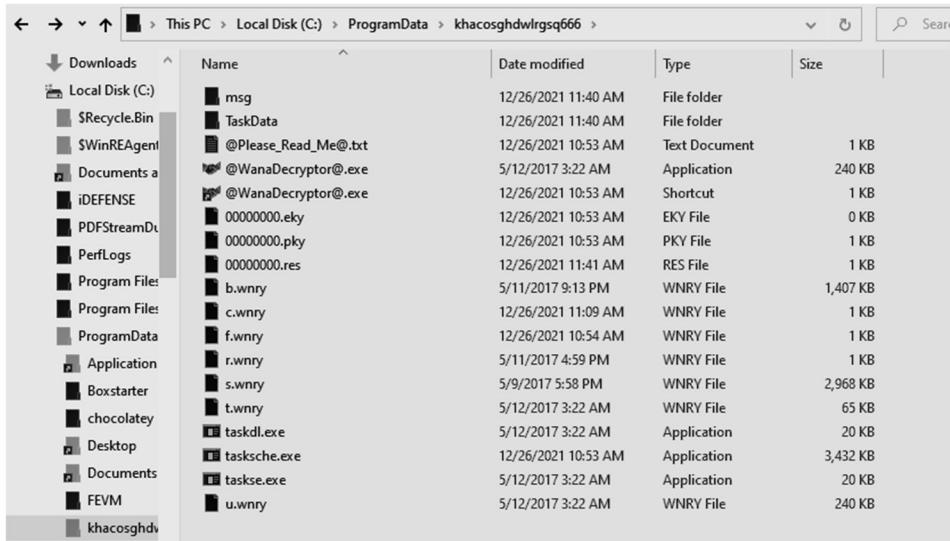
The initial executable that runs after a successful EternalBlue exploit, and failure to reach the callback URL. This executable is primarily responsible for the propagation as opposed to the encryption. After creating the staging directory and attempting to persist, wannacry.exe will proceed to search for other hosts on all connected networks using multiple threads of execution.

## tasksche.exe

The second stage executable that runs after it has been unpacked from wannacry.exe. This binary conducts most of the malicious operations on the host, primarily the encryption of the files on the compromised host.

## Additional Files

As mentioned earlier, a staging directory is created in C:\ProgramData once the call to the “kill switch” URL has failed. This directory contains a number of files outside of the two stages that are not explicitly additional stages for the malware.



The screenshot shows a Windows File Explorer window with the following details:

- Path:** This PC > Local Disk (C:) > ProgramData > khacosghdwrlgsq666
- File List:**

Name	Date modified	Type	Size
msg	12/26/2021 11:40 AM	File folder	
TaskData	12/26/2021 11:40 AM	File folder	
@Please_Read_Me@.txt	12/26/2021 10:53 AM	Text Document	1 KB
@WanaDecryptor@.exe	5/12/2017 3:22 AM	Application	240 KB
@WanaDecryptor@.exe	12/26/2021 10:53 AM	Shortcut	1 KB
0000000.eky	12/26/2021 10:53 AM	EKY File	0 KB
0000000.pkv	12/26/2021 10:53 AM	PKV File	1 KB
0000000.res	12/26/2021 11:41 AM	RES File	1 KB
b.wnry	5/11/2017 9:13 PM	WNRY File	1,407 KB
c.wnry	12/26/2021 11:09 AM	WNRY File	1 KB
f.wnry	12/26/2021 10:54 AM	WNRY File	1 KB
r.wnry	5/11/2017 4:59 PM	WNRY File	1 KB
s.wnry	5/9/2017 5:58 PM	WNRY File	2,968 KB
t.wnry	5/12/2017 3:22 AM	WNRY File	65 KB
taskdl.exe	5/12/2017 3:22 AM	Application	20 KB
tasksche.exe	12/26/2021 10:53 AM	Application	3,432 KB
taskse.exe	5/12/2017 3:22 AM	Application	20 KB
u.wnry	5/12/2017 3:22 AM	WNRY File	240 KB

Fig 2: Screenshot from the system explorer of the files in the “staging” directory.

Listed below are the files typically found in this directory and their function.

- msg/: The directory containing various translations of the ransom note
- TaskData/: Contains various files used in the Tor browser program
- 00000000.[eky | pky | res]: Purpose unknown; Seem to be various files with keys used in the encryption process
- b.wnry: The desktop background image used
- c.wnry: Contains data for certain “dark web” sites, a bitcoin wallet address, and a link to a Tor browser installation
- r.wnry: The README file to instruct victims how to decrypt their files
- s.wnry: Contains data with references to the Tor program
- t.wnry: Contains
- u.wnry: The Decryptor executable that shows the ransom note
- taskdl.exe: Locates files that have already been encrypted
- tasksche.exe: The second stage for the malware
- taskse.exe: Purpose unknown; seems to run the executable that gets passed in as an argument

# Basic Static Analysis

## Preliminary Information Gathering

### Architecture

```
C:\Users\sreisz\Desktop
λ file Ransomware.wannacry.exe.malz
Ransomware.wannacry.exe.malz: PE32 executable (GUI) Intel 80386, for MS Windows
```

### Hashes

Type	Hash
MD5	db349b97c37d22f5ea1d1841e3c89eb4
SHA256	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

## Strings

Below are annotated screenshots of interesting strings found in the sample.

```
--USERID__PLACEHOLDER__@  
h6agLCqPqVyXi2VSQ806Yb9ijBX54jY6KM+sz33NmS6TK8Xl0k920s0E0aaJ0V++wrR92ds1F0LB0+evLPj4  
sIvAjLvaLdgk8+Blnzs8PMa9bQ340J83nx1p4f+GLpbxUy...  
[trim]...vXbwtmRkgiGneql4mBYmKDzcXCkp/tjnL6/KriY81gMHN4G9ulMunxVyF8wybDcift0xtarjLXV  
RuC1Y7vzYaEuHT  
SMB3  
--TREEID__PLACEHOLDER__  
--USERID__PLACEHOLDER__@  
h54Wff9cGigWFEx92bzm0d0U0aZlMDdU2F4F2+6qn9/ZDSqJksnLIfbd0iMA3D+1qUTSrerHhgCcS2PibZuz  
q9y+eWL0zwmwXaWqkEMg2LU3HWJN4+Sf5DkSGjBmXQb0UQ...  
[trim]...czk25ArAzCQDX1MRxY20HuT3rhmyYLpiuJX/mu7wb6CGWZ4i6/eo1XB3sb3ucvGEzAheJm9zxN  
H3/tcqpc4MtJe/60AwTD+e362d6  
bbCUB+5x4jIXypy610lDcDWgbfIXcwcI02u15qZXg4cV/VjsDiEQARjmMebJBucJxC7HA9GSmUefyzAun9fL  
ULv3RbywhnNACbSX9hbRj/rxlAlfKv1cBRDwhcdL9p+vmw...  
[trim]...xLi7tEokjkrjiuxTJ7VOLMMoSqihIRgpTXkEvW4yy301fgQ+bAb0PNcCPaSxznfpGq9Rcq8uTkC  
gqDKEBujpjKKYi4BHd  
---
```

Interesting, long, encrypted(?) strings  
Repeats multiple times in the output

```
\%s\IPC$  
Microsoft Base Cryptographic Provider v1.0  
%d.%d.%d.%d
```

Potential calls to other SMB  
Shares on local network

```
mssecsvc2.0  
Microsoft Security Center (2.0) Service  
%s -m security
```

Further supporting service persistence  
hypothesis

```
C:\%s\queriuwjhrf  
C:\%s\%
```

Interesting path

WINDOWS

```
tasksche.exe  
CloseHandle  
WriteFile  
CreateFileA  
CreateProcessA
```

<http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com>

Potential callback URL



```
Microsoft Enhanced RSA and AES Cryptographic Provider
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
cmd.exe /c "%s"
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhwYgXeQepoHkHSQuy6NgaEb94
%s%d
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
t.wnry
icacls . /grant Everyone:F /T /C /Q
```

More encryption imports

```
cmd.exe /c "%s"
```

Command execution

```
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhwYgXeQepoHkHSQuy6NgaEb94
```

Interesting strings

```
%s%d
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
t.wnry
icacls . /grant Everyone:F /T /C /Q
```

Modified permissions

```
USER32.DLL
Windows 2000 2195
Windows 2000 5.0
\\172.16.99.5\IPC$
```

More shares, private IPs

```
Windows 2000 2195
Windows 2000 5.0
\\192.168.56.20\IPC$
```

```
kernel32.dll
WanaCrypt0r
```

Interesting string

```
Software\
.sqlite3
.sqlite3
.onetoc2
%$Intel
%$ProgramData
VS_VERSION_INFO
StringFileInfo
040904B0
CompanyName
Microsoft Corporation
FileDescription
DiskPart
FileVersion
6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName
diskpart.exe
```

False metadata?



## PEStudio

### PEStudio Indicators

			level
indicators (89)	indicator (89)	detail	
virustotal (error)	The file references string(s)	type: blacklist, count: 54	1
dos-header (64 bytes)	The file contains another file	signature: executable, location: .data, offset: 0x0000B020, size: 5263716	1
dos-stub (184 bytes)	The file contains another file	signature: executable, location: .data, offset: 0x0000F080, size: 5297524	1
rich-header (9)	The file contains another file	signature: executable, location: .data, offset: 0x000274BE, size: 159744	1
file-header (Nov.2010)	The size of a resource is suspicious	resource: R.1831	1
optional-header (GUI)	The size of a resource is suspicious	resource: R.1831	1
directories (3)	The file contains another file	signature: executable, location: .rsrc, offset: 0x000320A4, size: 3514368	1
sections (files)	The file references functions(s)	type: blacklist, count: 29	1
libraries (7) *	The file references a URL pattern	url: http://www.iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea.com	1
functions (91)	The file references file extensions like a Ransomware   Wiper	count: 146	1
exports (n/a)	The file references a string with a suspicious size	size: 2039 bytes	2
tls-callbacks (n/a)	The file references a string with a suspicious size	size: 1403 bytes	2

Fig. 3.1

- Suggests that there might be more stages to be unpacked
- “The file references file extensions like a Ransomware | Wiper” correlates with this being ransomware
- “The file references a string with a suspicious size” falls in line with some of the long, encrypted strings found earlier

### Version

pestudio 9.22 - Malware Initial Assessment - www.winitor.com		
file	settings	
indicators (89)	property	
virustotal (error)	value	
dos-header (64 bytes)	md5	<a href="#">1EBDC36976DD611E1A9E221A88E6858E</a>
dos-stub (184 bytes)	sha1	<a href="#">7B5A93CD7DB3DDC7FF48C6E3C7EEFCA46807462E</a>
rich-header (9)	sha256	<a href="#">2F3FC51546ADA848DFC8E775554C0DE3689D6FAE7BA4BF3D40E3C8DEC68B277B</a>
file-header (Nov.2010)	file-type	executable
optional-header (GUI)	date	empty
directories (3)	language	English-US
sections (files)	code-page	Unicode UTF-16, little endian
libraries (7) *	CompanyName	Microsoft Corporation
functions (91)	FileDescription	Microsoft® Disk Defragmenter
exports (n/a)	FileVersion	6.1.7601.17514 (win7sp1_ntm.101119-1850)
tls-callbacks (n/a)	InternalName	lhdfrgui.exe
.NET (n/a)	LegalCopyright	© Microsoft Corporation. All rights reserved.
resources (executable) *	OriginalFilename	<a href="#">lhdfrgui.exe</a>
abc strings (size)	ProductName	Microsoft® Windows® Operating System
	ProductVersion	6.1.7601.17514

Fig. 3.2

The executable seems to have metadata to masquerade as an official Microsoft product.



## Imports

pestudio 9.22 - Malware Initial Assessment - www.winitor.com

file settings about

Imports

c:\users\sreisz\desktop\ransomware.wannacry.exe

indicators (89)

dos-header (64 bytes)

dos-stub (184 bytes)

rich-header (9)

file-header (Nov.2010)

optional-header (GUI)

directories (3)

sections (files)

libraries (7) \*

functions (91)

exports (n/a)

tls-callbacks (n/a)

.NET (n/a)

resources (executable) \*

strings (size)

debug (n/a)

manifest (n/a)

version (lhdfregui.exe)

certificate (n/a)

overlay (n/a)

functions (91)	blacklist (29)	ordinal (13)	library (7)
TerminateThread	x	-	kernel32.dll
QueryPerformanceFrequency	x	-	kernel32.dll
StartServiceCtrlDispatcherA	x	-	advapi32.dll
ChangeServiceConfig2A	x	-	advapi32.dll
CreateServiceA	x	-	advapi32.dll
CryptGenRandom	x	-	advapi32.dll
CryptAcquireContextA	x	-	advapi32.dll
3 (closesocket)	x	x	ws2_32.dll
16 (recv)	x	x	ws2_32.dll
19 (send)	x	x	ws2_32.dll
8 (htonl)	x	x	ws2_32.dll
14 (ntohl)	x	x	ws2_32.dll
115 (WSAStartup)	x	x	ws2_32.dll
12 (inet_ntoa)	x	x	ws2_32.dll
10 (ioctlsocket)	x	x	ws2_32.dll
18 (select)	x	x	ws2_32.dll
9 (htons)	x	x	ws2_32.dll
23 (socket)	x	x	ws2_32.dll
4 (connect)	x	x	ws2_32.dll
11 (inet_addr)	x	x	ws2_32.dll
GetAdaptersInfo	x	-	iphlpapi.dll
InternetOpenA	x	-	wininet.dll
InternetOpenUrlA	x	-	wininet.dll
InternetCloseHandle	x	-	wininet.dll

Fig 3.3

- The calls related to Services potentially indicate some kind of service-based persistence
- CryptGenRandom and CryptAcquireContextA fall in line with this sample being ransomware. Additional calls to the cryptographic functions in the Windows API were also found
- InternetOpenA and other imports follow the typical setup for a malicious binary to access some web resource.

# Basic Dynamic Analysis

## Initial Detonation – With Internet Simulation

After running the sample with Administrator permissions, the following packets were captured using Wireshark, from the connected Remnux VM.

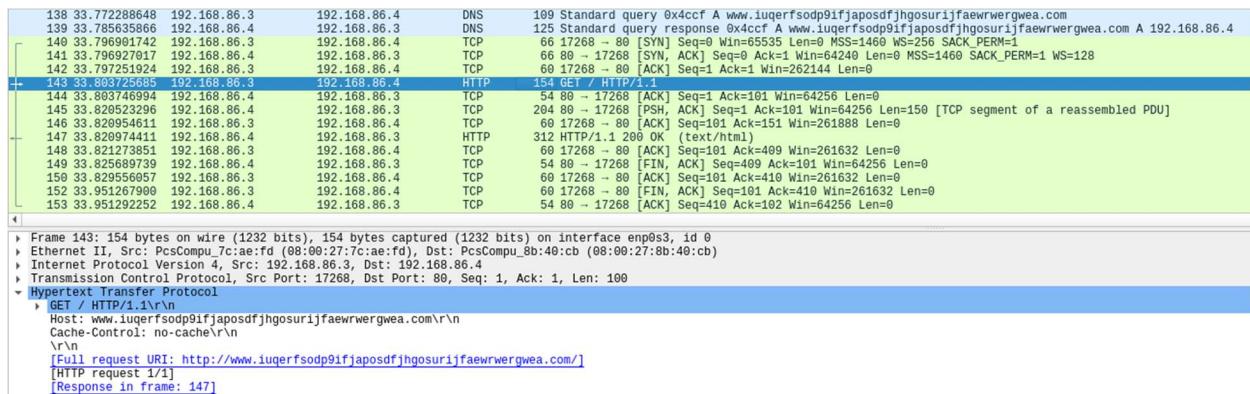


Fig 4.1: Packet capture showing the HTTP request to the callback URL.

However, monitoring on the host using procmon showed that the sample terminated execution after this point.

Description:	Microsoft® Disk Defragmenter		
Company:	Microsoft Corporation		
Path:	C:\Users\sreisz\Desktop\Ransomware.wannacry.exe		
Command:	"C:\Users\sreisz\Desktop\Ransomware.wannacry.exe"		
User:	DESKTOP-U6VBER0\sreisz		
PID:	4960	Started:	12/27/2021 10:17:10 AM
		Exited:	12/27/2021 10:17:11 AM

Fig 4.2: The sample's metadata claims to be an officially licensed Microsoft tool.

No additional artifacts were found on the system during initial detonation in the presence of an internet simulation.

## Initial Detonation – Without Internet Simulation

After disabling the internet simulation and running the sample again, DNS requests to the same URL are observed again. Since internet simulation is disabled, these requests fail.

No.	Time	Source	Destination	Protocol	Length	Info
99	6.788547	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com
100	6.788873	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
101	6.789020	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com
102	6.789216	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
103	6.789284	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com
104	6.789471	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
105	6.789634	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com
106	6.789810	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
107	6.790715	192.168.86.3	192.168.86.4	DNS	109	Standard query 0xd0b7 A www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com
108	6.790930	192.168.86.4	192.168.86.3	ICMP	137	Destination unreachable (Port unreachable)
190	11.309662	PcsCommu_7c:aef:fd	PcsCommu_8b:40:ch	ARP	42	Who has 192.168.86.4? Tell 192.168.86.3
> Internet Protocol Version 4, Src: 192.168.86.3, Dst: 192.168.86.4						
> User Datagram Protocol, Src Port: 59004, Dst Port: 53						
Domain Name System (query)						
Transaction ID: 0xd0b7						
> Flags: 0x0100 Standard query						
Questions: 1						
Answer RRs: 0						
Authority RRs: 0						
Additional RRs: 0						
< Queries						
> www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com: type A, class IN						

Fig 4.3: Failed requests to the callback URL.

However, the sample continues to execute. New files are observed on the Desktop.



Fig 4.4: Screenshot of the Desktop, where new files, such as @WanaDecryptor@.exe have been placed.

Using procmon, we note the creation of a new file, tasksche.exe, in C:\Windows. This file creates a new, hidden directory: C:\ProgramData\khacosghdwlrsgq666. The contents of this directory can be seen in Figure 2.

10:53...	tasksche.exe	720	CreateFile	C:\ProgramData	SUCCESS Desired Access: R...
10:53...	tasksche.exe	720	QueryBasicInfor...	C:\ProgramData	SUCCESS CreationTime: 12/7...
10:53...	tasksche.exe	720	CloseFile	C:\ProgramData	SUCCESS
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData	NAME COLLISION Desired Access: R...
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData	SUCCESS Desired Access: E...
10:53...	tasksche.exe	720	CloseFile	C:\Users\sreisz\Desktop	SUCCESS
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrsgq666	SUCCESS Desired Access: R...
10:53...	tasksche.exe	720	ReadFile	C:\\$Secure\\$SDH:\$INDEX_ALLOCATION	SUCCESS Desired Access: R...
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrsgq666	Offset: 73,728, Len...
10:53...	tasksche.exe	720	CloseFile	C:\ProgramData\khacosghdwlrsgq666	SUCCESS Desired Access: E...
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrsgq666\khacosghdwlrsgq666	SUCCESS
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrsgq666\khacosghdwlrsgq666	NAME NOT FOUND Desired Access: R...
10:53...	tasksche.exe	720	RegOpenKey	HKEY_LOCAL_MACHINE\Software\WOW6432Node\Policies\Microsoft\Windows\System	NAME NOT FOUND Desired Access: W...
10:53...	tasksche.exe	720	RegOpenKey	HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System	REPARSE Desired Access: Q...
10:53...	tasksche.exe	720	RegSelInfoKey	HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System	SUCCESS Desired Access: Q...
10:53...	tasksche.exe	720	RegQueryValue	HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System\CopyFileBufferedSynchronousIo	SUCCESS KeyInformation...
10:53...	tasksche.exe	720	RegCloseKey	HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\System	NAME NOT FOUND Length: 20
10:53...	tasksche.exe	720	CreateFile	C:\Windows\tasksche.exe	SUCCESS
10:53...	tasksche.exe	720	QueryAttributeT...	C:\Windows\tasksche.exe	SUCCESS Desired Access: G...
10:53...	tasksche.exe	720	CloseFile	C:\Windows\tasksche.exe	SUCCESS Attributes: SA, Rep...
10:53...	tasksche.exe	720	CreateFile	C:\Windows\tasksche.exe	SUCCESS Desired Access: G...
10:53...	tasksche.exe	720	QueryStandardI...	C:\Windows\tasksche.exe	SUCCESS AllocationSize: 3,5...
10:53...	tasksche.exe	720	QueryBasicInfor...	C:\Windows\tasksche.exe	SUCCESS CreationTime: 12/2...
10:53...	tasksche.exe	720	QueryStreamInf...	C:\Windows\tasksche.exe	SUCCESS 0 ::\$DATA
10:53...	tasksche.exe	720	QueryBasicInfor...	C:\Windows\tasksche.exe	SUCCESS CreationTime: 12/2...
10:53...	tasksche.exe	720	QueryEaInformat...	C:\Windows\tasksche.exe	SUCCESS EaSize: 0
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrsgq666\tasksche.exe	SUCCESS Desired Access: G...
10:53...	tasksche.exe	720	CloseFile	C:\ProgramData\khacosghdwlrsgq666\tasksche.exe	SUCCESS Desired Access: G...
10:53...	tasksche.exe	720	CreateFile	C:\ProgramData\khacosghdwlrsgq666\tasksche.exe	SUCCESS Desired Access: G...
<hr/>					
Ransomware.wannacry.exe	[Microsoft® Disk D...	C:\Users\sreisz\Des...	Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
cmd.exe (3992)	Windows Comm... C:\Windows\sy...	Microsoft Corporat...	NT AUTHORITY\...	cmd.exe /c "C:\ProgramData\khacosghdwlrsgq666\task...	12/26/2021 10:5...
tasksche.exe (6004)	DiskPart	C:\ProgramData\...	Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
attrib.exe (4384)	Attribute Utility	C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	n/a
Conhost.exe (526)	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
icacls.exe (1916)		C:\Windows\Syst...	Microsoft Corporat...	icacls . /grant Everyone:F /T /C /Q	12/26/2021 10:5...
Conhost.exe (561)	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
taskkill.exe (5848)	SQL Client Config...	C:\ProgramData\...	Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
cmd.exe (2944)	Windows Comm... C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	taskkill.exe	12/26/2021 10:5...
Conhost.exe (430)	Console Window ...	C:\Windows\Syst...	Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
cscript.exe (5132)	[Microsoft® Cons... C:\Windows\Sys...	Microsoft Corporat...	NT AUTHORITY\...	87231640544811.bat	12/26/2021 10:5...
			Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
			Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...
			Microsoft Corporat...	NT AUTHORITY\...	12/26/2021 10:5...

Fig 4.5: Procmon logs detailing the creation of the staging directory.

Following these events, three more immediate observations are made.

1. Files that are typically used to store data (e.g. .pdf, .jpeg, .txt) are encrypted and have the extension “.WNRY” appended to them.
2. On the network, a massive number of ARP packets<sup>1</sup> originate from the compromised host.
3. After some time, the desktop wallpaper is replaced with a note saying “Ooops, your important files are encrypted”. The @WanaDecyryptor@.exe pops up every minute or so, demanding a ransom payment of 300 USD in Bitcoin to the given wallet.

<sup>1</sup> In an actual, non-laboratory environment, these packets would likely be SMB connections as opposed to ARP requests. This is because one of the network interfaces on the Windows host is assigned a link-local address, and the sample attempts to find other hosts with link-local addresses, but there are none on the network.



password.txt.WNCRY - Notepad

File Edit Format View Help

WANACRY! [0~/h•÷Y01"=^ÈÖ€#ø°"CÀh\*] ;ÁþšH. iHýœ"||³³Ž¤||5£&ç" xG@B ^  
~Ä-6æÔ^Z d9||, Yµ ,%Ü0ç^| | !||jAŠOŠÍ®á,,

Ln 1, Col 1 100% Macintosh (CR) ANSI

99195	999.054318179	192.168.86.3	192.168.86.4	DNS
99196	999.054428828	192.168.86.3	192.168.86.4	DNS
99197	999.054577823	192.168.86.3	192.168.86.4	DNS
99198	999.067460723	PcsCompu_7c:ae:fd	Broadcast	ARP
99199	999.086580504	PcsCompu_7c:ae:fd	Broadcast	ARP
99200	999.086950844	192.168.86.3	192.168.86.4	DNS
99201	999.087388125	192.168.86.3	192.168.86.4	DNS
99202	999.087530678	192.168.86.3	192.168.86.4	DNS
99203	999.116444278	PcsCompu_7c:ae:fd	Broadcast	ARP
99204	999.116444653	PcsCompu_7c:ae:fd	Broadcast	ARP
99205	999.116713254	PcsCompu_7c:ae:fd	Broadcast	ARP
99206	999.116713392	PcsCompu_7c:ae:fd	Broadcast	ARP
99207	999.116713431	PcsCompu_7c:ae:fd	Broadcast	ARP
99208	999.116972087	PcsCompu_7c:ae:fd	Broadcast	ARP
99209	999.116972343	PcsCompu_7c:ae:fd	Broadcast	ARP
99210	999.117201678	PcsCompu_7c:ae:fd	Broadcast	ARP

Frame 99202: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu\_7c:ae:fd (08:00:27:7c:ae:fd), Dst: PcsCompu\_8b:40:cb (08:00:27:8b:40:cb)

Internet Protocol Version 4, Src: 192.168.86.3, Dst: 192.168.86.4

User Datagram Protocol, Src Port: 53, Dst Port: 53

Domain Name System (query)

Transaction ID: 0x990c

Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

Queries

26.216.254.169.in-addr.arpa: type PTR, class IN

[Retransmitted request. Original request in: 99124]

[Retransmission: True]

Recycle Bin PMAT-lab...

WanaDec...@WanaDec...

Oops, your important files are encrypted.

If you see this text, but don't see t

@Please\_Read\_Me@.txt - Notepad

File Edit Format View Help

Q: What's wrong with my files?

A: Oops, your important files are encrypted. It means you will not be able to access them anymore until they are decrypted. If you follow our instructions, we guarantee that you can decrypt all your files quickly and safely! Let's start decrypting!

Q: What do I do?

A: First, you need to pay service fees for the decryption. Please send \$300 worth of bitcoin to this bitcoin address: 115p7UMMmgd1pMvkphijcRdfJNXj6LrLn

Next, please find an application file named "WanaDecryptor.exe". It is the decrypt software. Run and follow the instructions! (You may need to disable your antivirus for a while.)

Q: How can I trust?

A: Don't worry about decryption. We will decrypt your files surely because nobody will trust us if we cheat users.

If you need our assistance, send a message by clicking <Contact Us> on the decryptor window.

Wana Decryptor 2.0

Payment will be raised on 12/29/2021 11:09:58

Time Left 02:23:57:43

Your files will be lost on 1/2/2022 11:09:58

Time Left 06:23:57:43

About Bitcoin How to buy Bitcoin? Contact Us

Send \$300 worth of bitcoin to this address:

115p7UMMmgd1pMvkphijcRdfJNXj6LrLn

Check Payment Decrypt

Evaluation or 86 days 11:21 AM 12/26/2021

Fig 4.6: Various screenshots showing the indicators: (top to bottom) an encrypted text file, packet capture with large volume of ARP requests, compromised host desktop

WannaCry Ransomware Cryptoworm  
Dec 2021  
v1.0

Aside from this, persistence was found in the Services<sup>2</sup>.

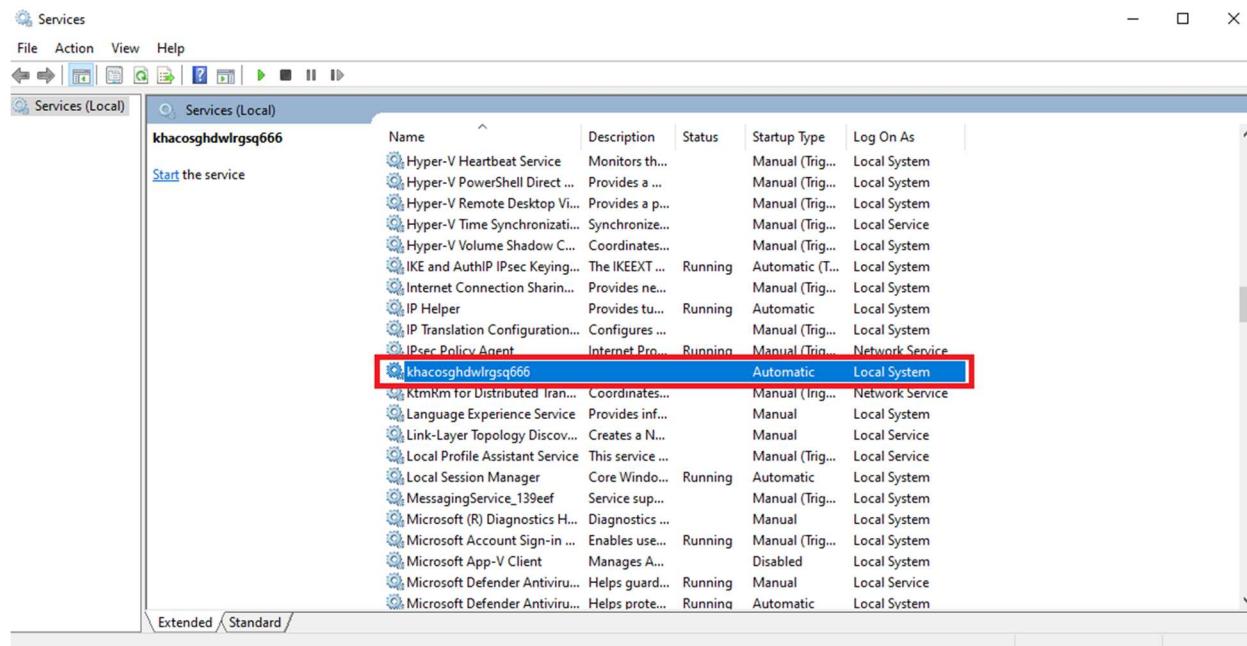


Fig 4.7: Screenshot from the Windows Services showing the creation of a new service.

To prevent victims from restoring their compromised machines to a previous state, the shadow copies used by the Volume Shadow Copy Service (VSS) are deleted.

```
cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignorefailures & bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet
!/?C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
wmic shadowcopy delete
taskse.exe C:\ProgramData\khacosghdwlrq666\@WanaDecryptor@.exe
"C:\ProgramData\khacosghdwlrq666\@WanaDecryptor@.exe"
cmd.exe /c reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "khacosghdwlrq666" /t REG_SZ /d "\"C:\ProgramData\khacosghdwlrq666\tasksche.exe\""/f
!/?C:\Windows\system32\conhost.exe 0xffffffff ForceV1
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "khacosghdwlrq666" /t REG_SZ /d "\"C:\ProgramData\khacosghdwlrq666\tasksche.exe\""/f
taskdl.exe
taskdl.exe
taskse.exe C:\ProgramData\khacosghdwlrq666\@WanaDecryptor@.exe
"C:\ProgramData\khacosghdwlrq666\@WanaDecryptor@.exe"
taskse.exe C:\ProgramData\khacosghdwlrq666\@WanaDecryptor@.exe
"C:\ProgramData\khacosghdwlrq666\@WanaDecryptor@.exe"
```

```
FLARE Fri 12/31/2021 17:23:18.86
C:\Users\sreisz>vssadmin list shadows
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.

No items found that satisfy the query.
```

Fig 4.8: Evidence of the wiped shadow copies.

<sup>2</sup> Although the sample attempted to modify keys in the registry to gain additional persistence, these changes were not actually seen in the Registry. Further investigation may be necessary.

## Interaction with Decryption Program

The three links at the bottom left of the “Wana DecryptOr 2.0” program redirect to google searches about each topic. This leaves the “Check Payment” and “Decrypt” functions.

Clicking the “Check Payment” button leads us to discover that the tcp/9050 port is open. It seems to function as a server that connects to remote addresses over HTTPS. When the button is clicked, a connection is made to it, and an error message is sent back.

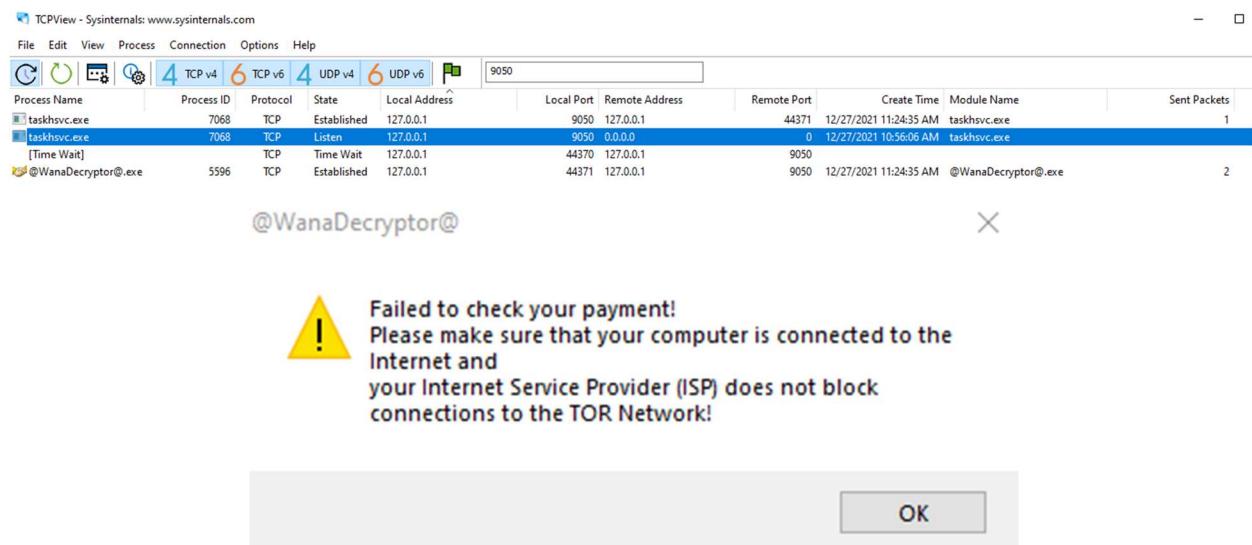
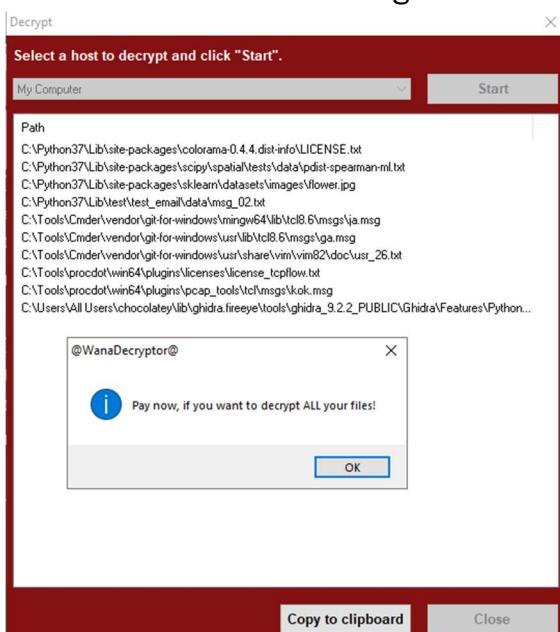


Fig 4.9: Evidence of the wiped shadow copies



The decryption program also gives victims a small “demo” of the decryption capability, likely to foster trust in the threat actors operating the malware. Clicking the “Decrypt” button shows a popup of the files that were decrypted, and then saying that payment was required to decrypt all files.

The full details of the encryption and decryption process have not been fully explored by this report at this time.

Fig 4.10: Decryption program.

# Advanced Static Analysis

Having had a better understanding of the functionality of the WannaCry ransomware, the Advanced Analysis phases had the following goals:

1. Understand the “kill switch” mechanism.
2. Find more information on how the ransomware propagates.
3. Find any additional functionality that was not previously observed.

Screenshots of larger disassembly segments can be found in the appendices.

The “kill switch” mechanism can be found in the disassembly of the main() function (Appendix B, Fig 8.2). The execution can be described as such:

1. The previously noted URL is moved into the ESI register.
2. Parameters are pushed onto the stack to call InternetOpenA, and the result of the call is moved into EAX and EDI.
3. The JNE instruction is called. If the call was unsuccessful, continue with the flow of the program. Otherwise, if the call was successful, exit the program.

The propagation mechanism seems to be within an “mssecservice2.0” service that is separate from the previously discussed service for persistence. Looking at the disassembly for more information about this proved to be inconclusive.

```

0x0040808d      ret 8
161: fcn.00408090 () ;
; var int32_t var_4h_3 @ esp+0xc
; var int32_t var_4h_2 @ esp+0x10
; var int32_t var_ch_2 @ esp+0x14
; var int32_t var_10h_2 @ esp+0x18
; var int32_t var_14h_2 @ esp+0x1c
; var char *lpServiceStartTable @ esp+0x20
; var int32_t var_ch @ esp+0x24
; var int32_t var_10h @ esp+0x28
; var int32_t var_14h @ esp+0x2c
0x00408090      sub esp, 0x10
0x00408093      push 0x104          ; 260
0x00408098      push 0x70f760
0x0040809d      push 0             ; HMODULE hModule
0x0040809f      call dword [GetModuleFileNameA] ; 0x40a06c ; DWORD GetModuleFileNameA(HMODULE hModule, LPSTR lpFilename, DWORD nSize)
0x004080a5      call dword [_p__argc] ; 0x40a12c
0x004080ab      cmp dword [eax], 2
0x004080ae      jge 0x4080b9
0x004080b0      call fcn.0040807f20
0x004080b5      add esp, 0x10
0x004080b8      ret
0x004080b9      push edi
0x004080ba      push 0xf003f        ; '?'
0x004080bf      push 0
0x004080c1      push 0             ; LPCSTR lpMachineName
0x004080c3      call dword [OpenSMangerA] ; 0x40a010 ; SC_HANDLE OpenSMangerA(LPCSTR lpMachineName, LPCSTR lpDatabaseName, DWORD dwDesiredAccess)
0x004080c9      mov eax, eax
0x004080cb      test edi, edi
0x004080cd      je 0x408101
0x004080cf      push ebx
0x004080d0      push esi
0x004080d1      push 0xf01ff
0x004080d2      push str.msecsvc2.0 ; 0x4312fc ; LPCSTR lpServiceName

```

*Fig 5: Disassembled code that creates the “mssecservice2.0”*

Since encryption was observed in the tasksche.exe binary, Advanced Static Analysis of the initial stage will not yield much more information.



# Advanced Dynamic Analysis

A hypothesis was previously made that a successful callback to the URL (`hxxp://www.iuquerfsodp9ifjaposdfjhgosurijfaewrwegwea.com`) would cause the sample to stop execution. This can be tested by putting the ransomware through a debugger. After starting up an internet simulation, we can step through the executable until we hit the JNE instruction that would normally exit out of the routine.

*Fig 6.1: The debugger just before the jump should be followed*

Using the debugger, we can set the zero flag to the Boolean value 1, such that we prevent the exit subroutine. Taking another step in the debugger proves our hypothesis, as the jump was not taken, and we can continue execution.

Fig 6.2: The result of switching the Zero Flag from 0 to 1



# Indicators of Compromise

## Network Indicators

- Calls to the hxxp://www.iuquerfsodp9ifjaposdfjhgosurijfaewrwegwae.com URL. Note that this URL has since been registered in DNS records to act as a sinkhole.

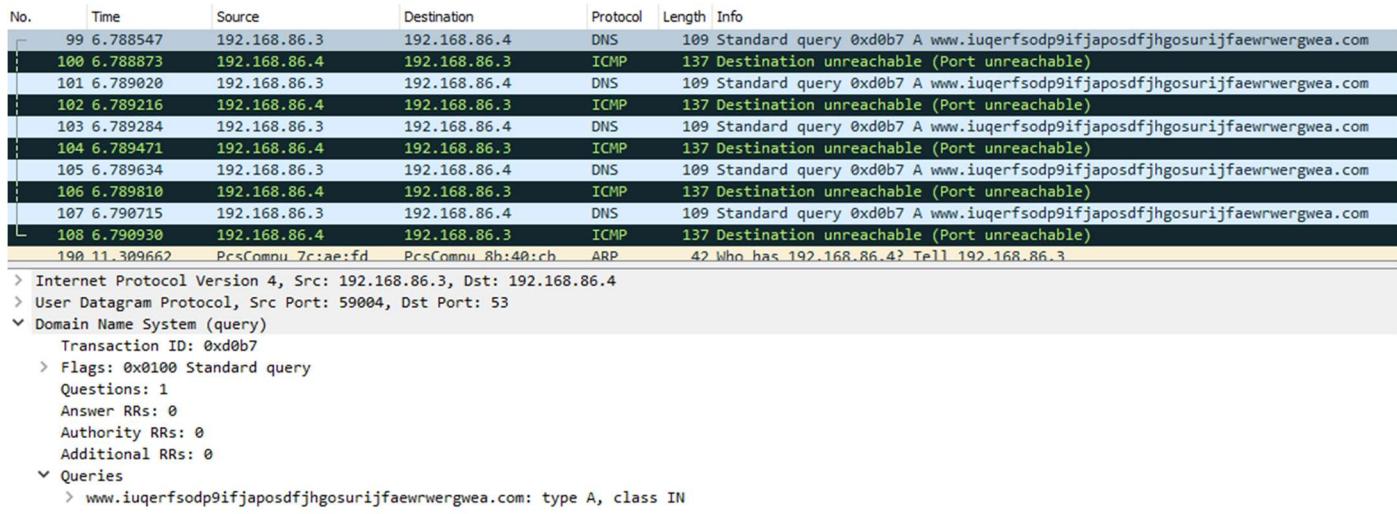


Fig 7.1: Wireshark Packet Capture of initial callback

- Flood of ARP and/or SMB packets leaving the compromised host.

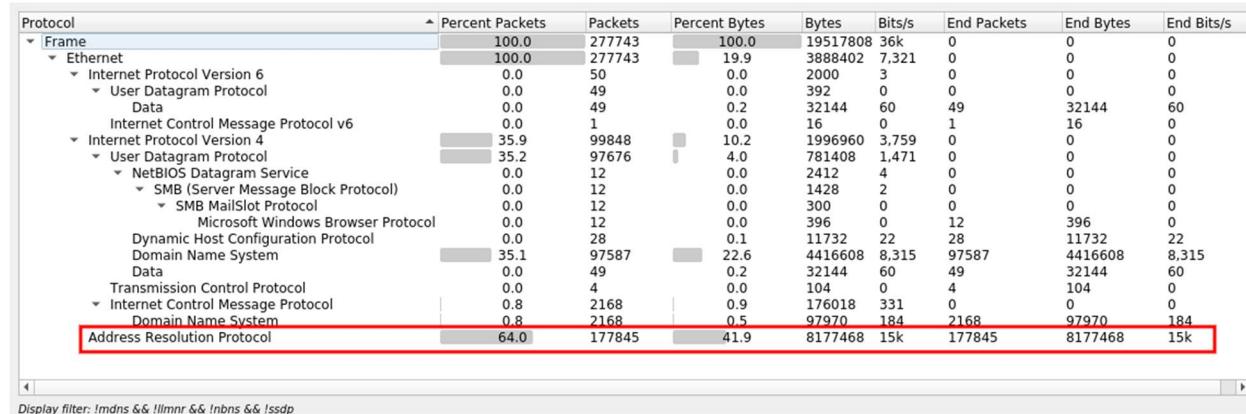
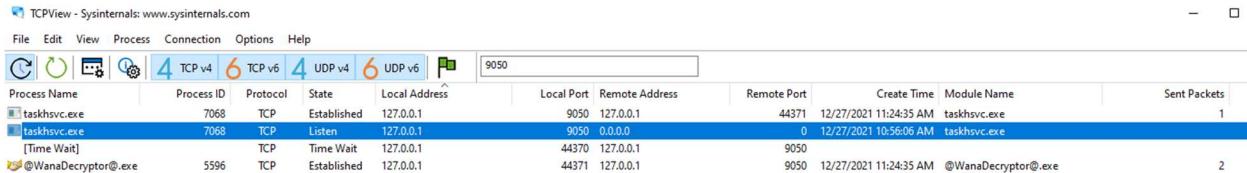


Fig 7.2: Wireshark statistics output noting that 64% of protocols were ARP

3. A process, taskhsvc.exe, listening on port tcp/9050. Attempts to make connections with non-private IP addresses may also be occurring.

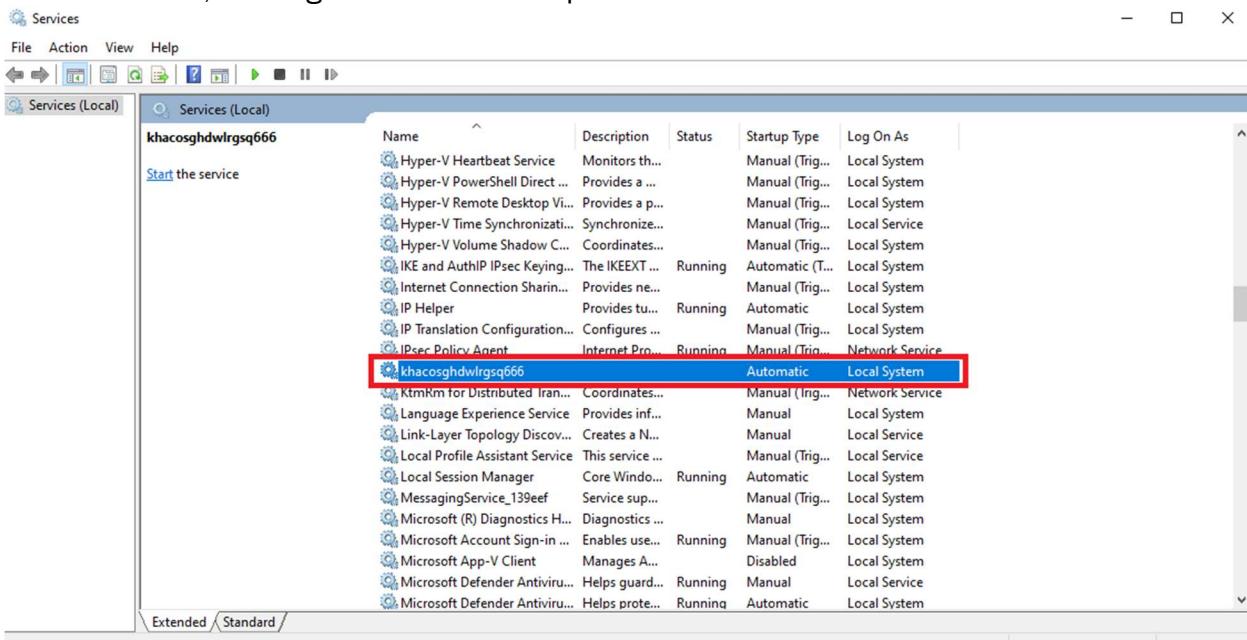


A screenshot of the TCPView application from Sysinternals. The interface shows network connections across four tabs: TCP v4, TCP v6, UDP v4, and UDP v6. The TCP v4 tab is selected, showing a list of connections. One connection is highlighted in blue: taskhsvc.exe (Process ID 7068) has a TCP connection established to 127.0.0.1 on port 9050. Another connection from taskhsvc.exe (Process ID 7068) is in a 'Time Wait' state, attempting to connect to 127.0.0.1 on port 9050. A third connection from @WanaDecryptor@.exe (Process ID 5596) is established to 127.0.0.1 on port 9050.

*Fig 7.3: Wireshark Packet Capture of initial callback*

## Host-based Indicators

1. The directory shown Fig 2 is created, containing a similar array of files. Note that the name of the directory may be different from host to host, but it will be hidden.
2. The existence of a new service, with the same name as the hidden directory, will be created, which gives tasksche.exe persistence



A screenshot of the Windows Services snap-in. The left pane shows the 'Services (Local)' list. A new service entry, 'khacosghdwlrq666', is highlighted with a red box. It has the following details:

Name	Description	Status	Startup Type	Log On As
khacosghdwlrq666		Running	Automatic (T...)	Local System

*Fig 7.4: Screenshot of the newly created, malicious service*

3. A visibly changed desktop, with the WannaDecryptOr ransom note periodically appearing at the top of a user's view (Shown in Fig. 4.6).
4. Shadowed copies of the filesystem are wiped from the system (Shown in Fig. 4.9).

## Rules & Signatures

A full set of YARA rules is included in Appendix A.

As the IOCs from the previous section have shown, this ransomware sample operates very overtly, and as such, has very apparent signatures.

- The ransomware makes it obvious what its name is. This will show up in various strings throughout the binary, such as “WANACRY!”, “wnry”, “WanaCryptOr”, etc.
- The presence of Bitcoin wallet addresses in the strings, while not necessarily an indicator of WannaCry, is reason to be suspicious.
  - 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
  - 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
  - 13AM4VW2dhwYgXeQepoHkHSQuy6NgaEb94
- Callbacks to URLs like the one discussed in this report gives high confidence that a sample may be WannaCry. Since the original attacks in 2017, many of these URLs have since been sinkholed, but source code could easily be modified.
- High volumes of SMB traffic and/or ARP requests originating from one or more hosts
- Anything to do with the Indicators of Compromise referenced throughout this document

# Appendices

## A. Yara Rules

Full repository located at: <http://github.com/An00bRektn/malware-analysis-reports>

```

rule Ransomware_Wannacry {
  meta:
    last_updated = "2021-12-30"
    author = "An00bRektn"
    description = "A simple yara rule to detect WannaCry"

  strings:
    $PE_magic_byte = "MZ"
    $x1 = "__TREEID__PLACEHOLDER__"
    $x2 = "__USERID__PLACEHOLDER__"
    $x3 = "wnry"

    // Catch hardcoded SMB syntax
    $s1 = /smb: \\\\(\d{1,3}\.){3}\d{1,3}\\IPC\$ /
    $s2 = "\\\%s\\IPC$" fullword ascii

    // Note that these three probably change between variations
    $s3 = "WANACRY!" fullword ascii
    $s4 = "WanaCrypt0r" fullword ascii
    $s5 = "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwegwea.com" ascii

  condition:
    $PE_magic_byte at 0 and
    ($x1 and $x2) and ($s1 or $s2) and ($s3 or $s4) or $x3 or $s5
}
  
```

*Fig 8.1: General YARA rule to identify WannaCry*



## B. Disassembled Code Snippets

```
[0x00408148]
139: int main (int argc, char **argv, char **envp);
; var int32_t var_14h @ esp+0x28
; var int32_t var_8h @ esp+0x3c
; var int32_t var_41h @ esp+0x75
; var int32_t var_45h @ esp+0x79
; var int32_t var_49h @ esp+0x7d
; var int32_t var_4dh @ esp+0x81
; var int32_t var_51h @ esp+0x85
; var int32_t var_55h @ esp+0x89
; var int32_t var_6bh @ esp+0x8b
sub    esp, 0x50
push   esi
push   edi
mov    ecx, 0xe           ; 14
mov    esi, str.http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwengvea.com ; 0x4313d0
lea    edi, [var_8h]
xor    eax, eax
rep    movsd dword es:[edi], dword ptr [esi]
mouzb byte es:[edi], byte ptr [esi]
mov    dword [var_41h], eax
mov    dword [var_45h], eax
mov    dword [var_49h], eax
mov    dword [var_4dh], eax
mov    dword [var_51h], eax
mov    word [var_55h], ax
push   eax
push   eax
push   eax
push   1                  ; 1
push   eax
mov    byte [var_6bh], al
call   dword [InternetOpenA] ; 0x40a134
push   0
push   0x84000000
push   0
lea    ecx, [var_14h]
mov    esi, eax
push   0
push   ecx
push   esi
call   dword [InternetOpenUrlA] ; 0x40a138
mov    edi, eax
push   esi
mov    esi, dword [InternetCloseHandle] ; 0x40a13c
test   edi, edi
jne    0x4081bc
```

```
[0x004081a7]
call  esi
push  0
call  esi
call  fcn.00408090
pop   edi
xor   eax, eax
pop   esi
add   esp, 0x50
ret   0x10
```

```
[0x004081bc]
call  esi
push  edi
call  esi
pop   edi
xor   eax, eax
pop   esi
add   esp, 0x50
ret   0x10
```

Fig 8.2: Disassembled main function in Cutter.